

|  |  |
| --- | --- |
| 基调听云上线手册 | |
| 手册版本 | V1.1 |

前言

**文档修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **标题** | -基调听云上线手册 | | | |
| **项目名称** |  | | | |
| **类别** | 管理文档 □ 设计方案 □ 实施文档 □  配置文档 □ 测试文档 □ 其他 □ | | | |
| **创建日期** | 2022/03/30 | | | |
| **一线人员** | 刘孟凯 | | | |
|  |  | | | |
| **版本号** | **日期** | **修改人** | **审阅人** | **摘要** |
| V1.0 | 2022/03/30 | 刘孟凯 |  | 编写 |
| V1.0 | 2022/03/30 | 刘孟凯 |  | 修改端口清单 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

版权说明

本文档中出现的任何文字叙述、文档格式、插图、照片、方法、代码等内容，除由特别注明，版权均属于以基调网络公司所有，受到有关产权及版权法保护。任何个人、机构未经基调网络公司的书面授权许可，不得以任何方式复制或引用本文档的任何片断。

目录

1 服务器配置需求说明 4

2 组件清单 6

2.1 全部组件清单 6

2.2 第三方组件版本清单 8

3 组件端口清单 9

4 系统架构说明 14

4.1 网络拓扑图 14

4.2 对外网络策略 15

4.3 逻辑架构图 16

5 环境说明 17

5.1 操作系统 17

5.2 基础环境 17

6 软件安装与配置 18

6.1 机器初始化 18

6.2 相关安装配置修改 19

6.3 执行安装 20

6.4 输出日志说明 20

概述

# 服务器配置需求说明

****

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 配置 | 说明 |
| 1 | common-01 | 8C 16G 200G | 报表、授权等公共组件+数据接收NGINX |
| 2 | common-02 | 8C 16G 200G |
| 3 | collector-01 | 4C 8G 50G | APM数据收集器 |
| 4 | collector-02 | 4C 8G 50G |
| 5 | apm-01 | 8C 16G 200G | APM后端监控数据处理程序 |
| 6 | apm-02 | 8C 16G 200G |
| 7 | app-01 | 8C 16G 200G | APP监控数据处理程序 |
| 8 | app-02 | 8C 16G 200G |
| 9 | mysql/redis-master | 4C 8G 200G | 配置数据库&缓存服务器 |
| 10 | mysql/redis-slave | 4C 8G 200G |
| 11 | es/nbfs-01 | 8C 16G 500G | TRACE类型数据数据库 |
| 12 | es/nbfs-02 | 8C 16G 500G |
| 13 | es/nbfs-03 | 8C 16G 500G |
| 14 | kafak-01 | 4C 8G 300G | 消息队列服务器 |
| 15 | kafak-02 | 4C 8G 300G |
| 16 | kafak-03 | 4C 8G 300G |
| 17 | druid-01 | 8C 16G 500G | 指标类型数据数据库 |
| 18 | druid-02 | 8C 16G 500G |
| 19 | druid-03 | 8C 16G 500G |
| 20 | nginx-01 | 4C 8G 50G | 页面查看转发服务器 |
| 21 | nginx-02 | 4C 8G 50G | APP数据接收转发服务器 |
| 22 | nginx-03 | 4C 8G 50G | APP数据接收转发服务器 |

# 组件清单

## 全部组件清单

|  |  |
| --- | --- |
| **组件名称** | **组件描述** |
| base-druid | 基础组件 druid 时序数据库 |
| base-elasticsearch | 基础组件 elasticsearch 搜索数据库 |
| base-hadoop | 基础组件 hadoop 大数据处理/分布式存储 |
| base-kafka | 基础组件 kafka 分布式发布订阅消息系统 |
| base-mysql | 基础组件 mysql 数据库（配置数据） |
| base-nacos-server | 基础组件nacos配置中心服务 |
| base-nginx | 基础组件nginx静态web服务/反向代理服务/负载均衡服务 |
| base-redis | 基础组件redis数据缓存服务 |
| base-zookeeper | 基础组件zookeeper分布式应用程序协调服务 |
| common-api-gateway | common组件路由转发服务 |
| common-auth-api | common组件auth认证服务 |
| common-cluster-server | common组件url聚合等服务 |
| common-es-data-ingestion | common组件es数据入库服务 |
| common-ipdb-server | common组件ipdb服务 |
| common-license-server | common组件授权服务 |
| common-nbfs-server | common组件nbfs服务（类对象存储服务） |
| common-openresty | common组件openresty服务 |
| common-platform-api | common组件报表平台api |
| common-platform-web | common组件报表静态页面 |
| common-redirect-server | common组件redirect服务 |
| common-notification-api | common组件告警推送 |
| apm-api | apm组件报表查询api |
| apm-config | apm组件报表配置api |
| apm-dc-frontend | apm组件数据接收服务 |
| apm-dc-wrap | apm组件数据处理服务 |
| apm-health | apm组件健康度服务 |
| apm-web | apm组件报表静态页面 |
| app-api | app组件报表查询api |
| app-dc-frontend | app组件数据接收服务 |
| app-dc-wrap | app组件数据处理服务 |
| app-deobfuscator-dubbo | app组件提供dubbo服务 |
| app-statistics-elasticjob | app组件提供dubbo服务 |
| app-url-clustering-dubbo | app组件提供dubbo服务 |
| app-web | app组件报表静态页面 |
| browser-api | browser组件报表查询api |
| browser-url-normalize | browser组件url解析服务 |
| browser-web | browser组件报表静态页面 |
| browser-wrap | browser组件数据处理服 |
| infra-api | infra组件报表查询api |
| infra-dc-wrap | infra组件数据接收处理服务 |
| infra-metrics-correlator-aggregation | infra组件 |
| infra-metrics-correlator-compression | infra组件 |
| infra-web | infra组件报表静态页面 |
| alarm-api | alarm组件报表查询api |
| alarm-baseline | alarm组件后端基线服务 |
| alarm-dc | alarm组件后端数据处理服务 |
| alarm-timeseries-training | alarm组件后端时间序列训练服务 |
| alarm-web | alarm组件报表静态页面 |
| mp-web | mp组件报表静态页面 |
| mp-api | mp组件报表查询api |
| mp-wrap | mp组件数据处理服务 |
| mp-url-normalize-service | mp组件提供dubbo服务 |
| mp-url-normalize | mp组件提供dubbo服务 |
| visboard-sense-mock | visboard组件 |
| visboard-sense-api | visboard组件报表查询api |
| visboard-dashboard-api | visboard组件报表查询api |
| visboard-data-report-api | visboard组件报表查询api |
| visboard-sense-web | visboard组件报表静态页面 |
| bpi-dc-wrap | bpi组件后端数据处理服务 |
| bpi-api | bpi组件报表查询api |
| bpi-task | bpi组件后端数据处理服务 |
| bpi-grafana | bpi组件报表dashboard |
| bpi-adhoc-store | bpi组件后端数据处理服务 |
| bpi-adhoc-olap | bpi组件后端数据处理服务 |
| neuralert-api | neuralert组件报表查询api |
| neuralert-dc | neuralert组件数据接收处理服务 |

## 第三方组件版本清单

|  |  |  |  |
| --- | --- | --- | --- |
| 组件名称 | 版本信息 | License 信息 | 官网 |
| base-java | openjdk version "1.8.0\_232" | The GNU General Public License (GPL) | [https://www.azul.com/](https://www.azul.com/" \o "https://www.azul.com/) |
| base-nodejs | v14.18.1 |  |  |
| base-mysql | 8.0.26 | GNU General Public License (GPLv2) | [https://nodejs.org/](https://nodejs.org/" \o "https://nodejs.org/) |
| base-nginx | nginx version: nginx/1.18.0 |  | [http://nginx.org/](http://nginx.org/" \o "http://nginx.org/) |
| base-nacos-server | 1.4.2 | Apache-2.0 License | [https://nacos.io/](https://nacos.io/" \o "https://nacos.io/) |
| base-redis | 5.0.12 | BSD-3-Clause License | [https://redis.io/](https://redis.io/" \o "https://redis.io/) |
| base-zookeeper | 3.5.6 | Apache-2.0 License | [https://zookeeper.apache.org/](https://zookeeper.apache.org/" \o "https://zookeeper.apache.org/) |
| base-kafka | 2.12-2.3.1 | Apache-2.0 License | [https://kafka.apache.org/](https://kafka.apache.org/" \o "https://kafka.apache.org/) |
| base-elasticsearch | 7.4.2 | Elastic License 2.0 | [https://www.elastic.co/](https://www.elastic.co/" \o "https://www.elastic.co/) |
| base-hadoop | 2.10.1 | Apache-2.0 License | [https://hadoop.apache.org/](https://hadoop.apache.org/" \o "https://hadoop.apache.org/) |
| base-druid | 0.22.0 | Apache-2.0 License | [https://druid.apache.org/](https://druid.apache.org/" \o "https://druid.apache.org/) |

# 组件端口清单

全部组件端口清单

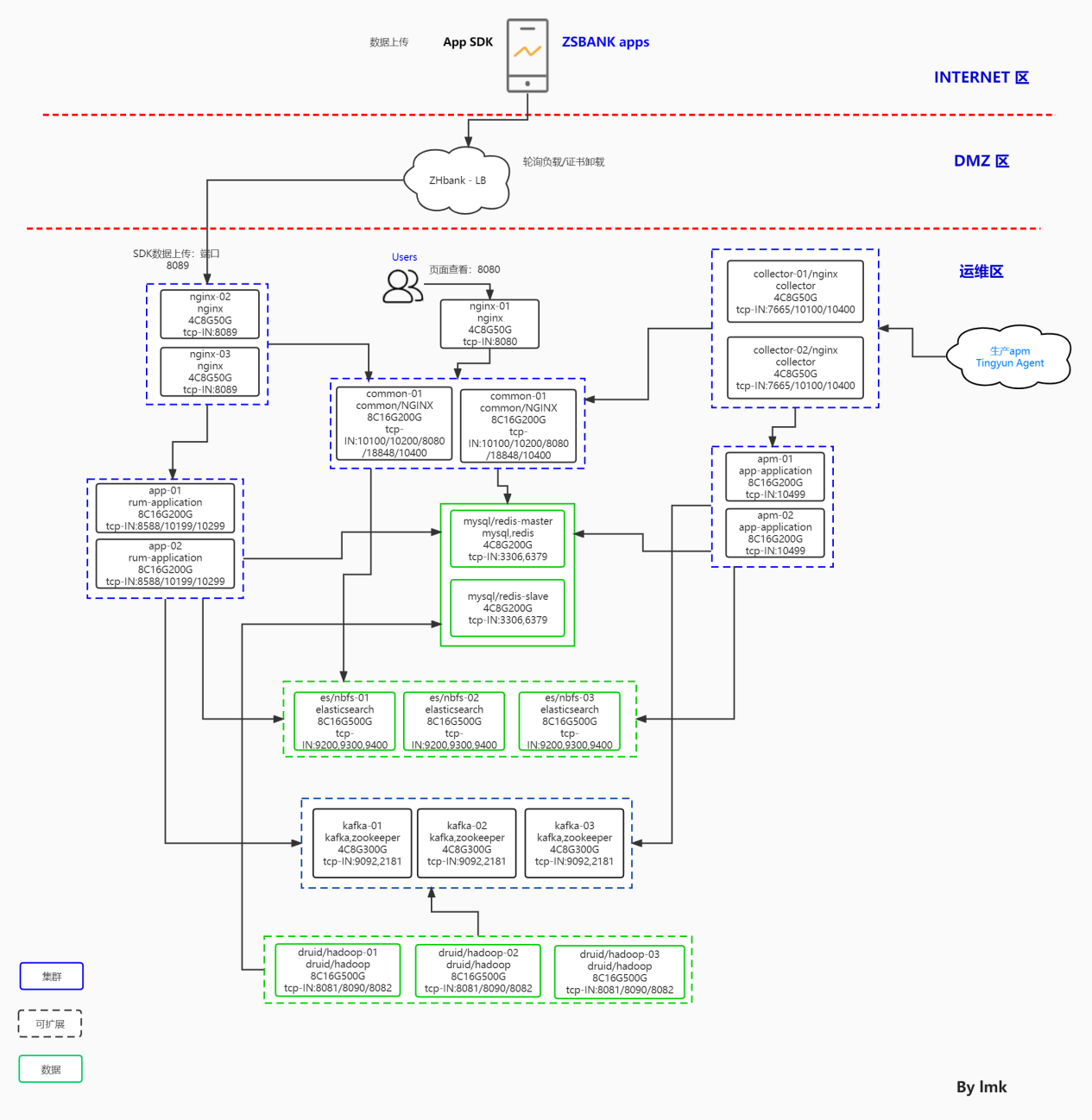


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **base组件** |  |  |  |  |
| **application\_name** |  | **application\_port** | **monitor\_port** | **others** |
| base-java |  |  |  |  |
| base-mysql |  | 3306 |  |  |
| base-mysql-slave |
| base-nginx |  | 8080 |  |  |
| base-nacos-server |  | 8848 | 28848 |  |
| base-redis |  | 6379 |  | redis-sentinel: 26379 |
| base-redis-slave |
| base-zookeeper |  | 2181 2888 3888 |  |  |
| base-kafka |  | 9092 |  |  |
| base-elasticsearch |  | 9200 9300 9400 |  |  |
| base-hadoop | namenode | 8020 |  | nn-web: 50070 |
| zkfc | 8019 |  |  |
| datanode | 50010 |  | dn-web: 50075 |
| dn-ipc: 8010 |
| journalnode | 8485 |  | jn-web: 8480 |
| base-druid | coordinator | 8081 | 28081 |  |
| overlord | 8090 | 28090 |  |
| broker | 8082 | 28082 |  |
| historical | 8083 | 28083 |  |
| indexer | 8091 | 28091 |  |
| router | 8888 | 置空 | 临时启动的查询服务，不做监控 |
| **common 组件** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| common-platform-web |  |  |  | web 项目 |
| common-doc-web |  |  |  |  |
| common-static-web |  |  |  |  |
| common-menu-i18n |  |  |  | web 项目 |
| common-es-data-ingestion |  | 20108 |  | springboot 项目 |
| common-license-server | 10107 | 20107 |  | springboot 项目 |
| common-nbfs-server | 10106 | 20106 | http-port: 10197 | spring 项目 |
| grpc-port: 10198 |
| kryolz4-port: 10195 |
| common-ipdb-server | 10105 | 20105 | http-port: 10196 | spring 项目 |
| common-cluster-server | 10104 | 20104 |  | spring 项目 |
| common-platform-api | 10103 | 20103 |  | springboot 项目 |
| common-auth-api | 10102 | 20102 |  | springboot 项目 |
| common-api-gateway | 10101 | 20101 |  | springboot 项目 |
| common-redirect-server | 10199 | 20199 | nginx-addr：10100 | springboot 项目 |
| common-openresty | 8588 |  |  | 开源 openresty，暂无监控 |
| common-notification-api | 10109 | 20109 |  | springboot 项目 |
| common-js-config | 10110 | 20110 |  | springboot 项目 |
| common-dashboard-api | 10111 | 20111 |  | springboot 项目 |
| common-source-map-api | 10112 | 20112 |  | springboot 项目 |
| common-source-map-parser | 10113 |  |  | nodejs 项目 |
| **app 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| app-web |  |  |  | web 项目 |
| app-api | 10201 | 20201 |  | springboot 项目 |
| app-dc-frontend | 10299 | 20299 | nginx-addr: 10200 | springboot 项目 |
| app-dc-wrap | 10202 | 20202 |  | springboot 项目 |
| app-deobfuscator-dubbo | 10204 | 20204 |  | springboot 项目 |
| app-statistics-elasticjob | 10205 | 20205 |  | springboot 项目 |
| app-url-clustering-dubbo | 10206 | 20206 |  | springboot 项目 |
| **apm 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| apm-web |  |  |  | web 项目 |
| apm-api | 10401 | 20401 |  | springboot 项目 |
| apm-config | 10402 | 20402 |  | springboot 项目 |
| apm-dc-frontend | 10499 | 20499 | nginx-addr: 10400 | springboot 项目 |
| apm-dc-wrap | 10403 | 20403 |  | springboot 项目 |
| apm-health | 10405 | 20405 |  | spring 项目 |
| apm-dynamic-baseline | 10409 | 20409 |  | spring 项目 |
| **browser 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| browser-web |  |  |  | web 项目 |
| browser-api | 10600 | 20600 |  | springboot 项目 |
| browser-config | 10601 | 20601 |  | springboot 项目 |
| browser-wrap | 10602 | 20602 |  | springboot 项目 |
| browser-url-normalize | 10603 | 20603 |  | springboot 项目 |
| browser-lighthouse | 10604 |  |  | node.js服务 |
| **mp 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| mp-web |  |  |  | web 项目 |
| mp-api | 10800 | 20800 |  | springboot 项目 |
| mp-config | 10801 | 20801 |  | springboot 项目 |
| mp-wrap | 10802 | 20802 |  | springboot 项目 |
| mp-url-normalize-service | 10803 | 20803 |  | springboot 项目 |
| mp-url-normalize | 10804 | 20804 |  | springboot 项目 |
| **infra 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| infra-web |  |  |  | web 项目 |
| infra-api | 11001 | 21001 |  | springboot 项目 |
| infra-dc-wrap | 11099 | 21099 | nginx-infra: 11000 | springboot 项目 |
| infra-timeseries-training | 11002 | 21002 |  | spring 项目 |
| infra-metrics-correlator-compression | 11003 | 21003 |  | spring 项目 |
| infra-metrics-correlator-aggregation | 11004 | 21004 |  | spring 项目 |
| **visboard 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| visboard-web |  |  |  | web 项目 |
| visboard-sense-web |  |  |  | web 项目 |
| visboard-sense-mock | 11200 | 21200 |  | nodejs mock 项目 |
| visboard-sense-api | 11201 | 21201 |  | springboot 项目 |
| visboard-data-report-api | 11203 | 21202 |  | springboot 项目 |
| **bpi 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| bpi-web |  |  |  | web 项目 |
| bpi-dc-wrap | 11400 | 21400 |  | springboot 项目 |
| bpi-api | 11401 | 21401 |  | springboot 项目 |
| bpi-task | 11402 | 21402 |  | springboot 项目 |
| bpi-grafana | 11403 | 21403 |  | golang 项目（二进制可执行文件） |
| bpi-adhoc-store | 11404 | 21404 |  | spring 项目 |
| bpi-adhoc-olap | 11405 | 21405 |  | spring 项目 |
| bpi-data-integration | 11406 | 21406 |  | springboot项目 |
| bcs-cmdb-api | 11407 | 21407 |  | springboot项目 |
| bpi-adhoc | 11408 | 21408 |  | spring 项目 |
| **neuralert 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| alarm-web |  |  |  | web 项目 |
| alarm-api | 11602 | 21602 |  | springboot 项目 |
| alarm-dc | 11603 | 21603 | 此项目提供了 sms/email 的 api | springboot 项目 |
| alarm-baseline | 11604 | 21604 |  | spring 项目 |
| alarm-timeseries-training | 11605 | 21605 |  | spring 项目 |
| neuralert-api | 11601 | 21601 |  | springboot 项目 |
| neuralert-dc | 11699 | 21699 | nginx-neuralert: 11600 | spring 项目 |
| backend-port:11698 |
| **metrics 产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| metrics-web |  |  |  | web 项目 |
| metrics-api | 11701 | 21701 |  | springboot 项目 |
| **rum产品线** |  |  |  |  |
| **application\_name** | **application\_port** | **monitor\_port** | **others** | **备注** |
| rum-kafka-sync |  |  |  |  |
| rum-sgcc-track-server | 11802 | 21802 |  | springboot 项目 |
| **自监控** |  |  |  |  |
| node\_exporter | 9100 |  |  |  |
| process\_exporter | 9256 |  |  |  |
| box\_exporter | 9078 |  |  |  |
| elasticsearch\_exporter | 9114 |  |  |  |
| kafka\_exporter | 9308 |  |  |  |
| druid\_exporter | 19215 |  |  |  |

# 系统架构说明

## 网络拓扑图



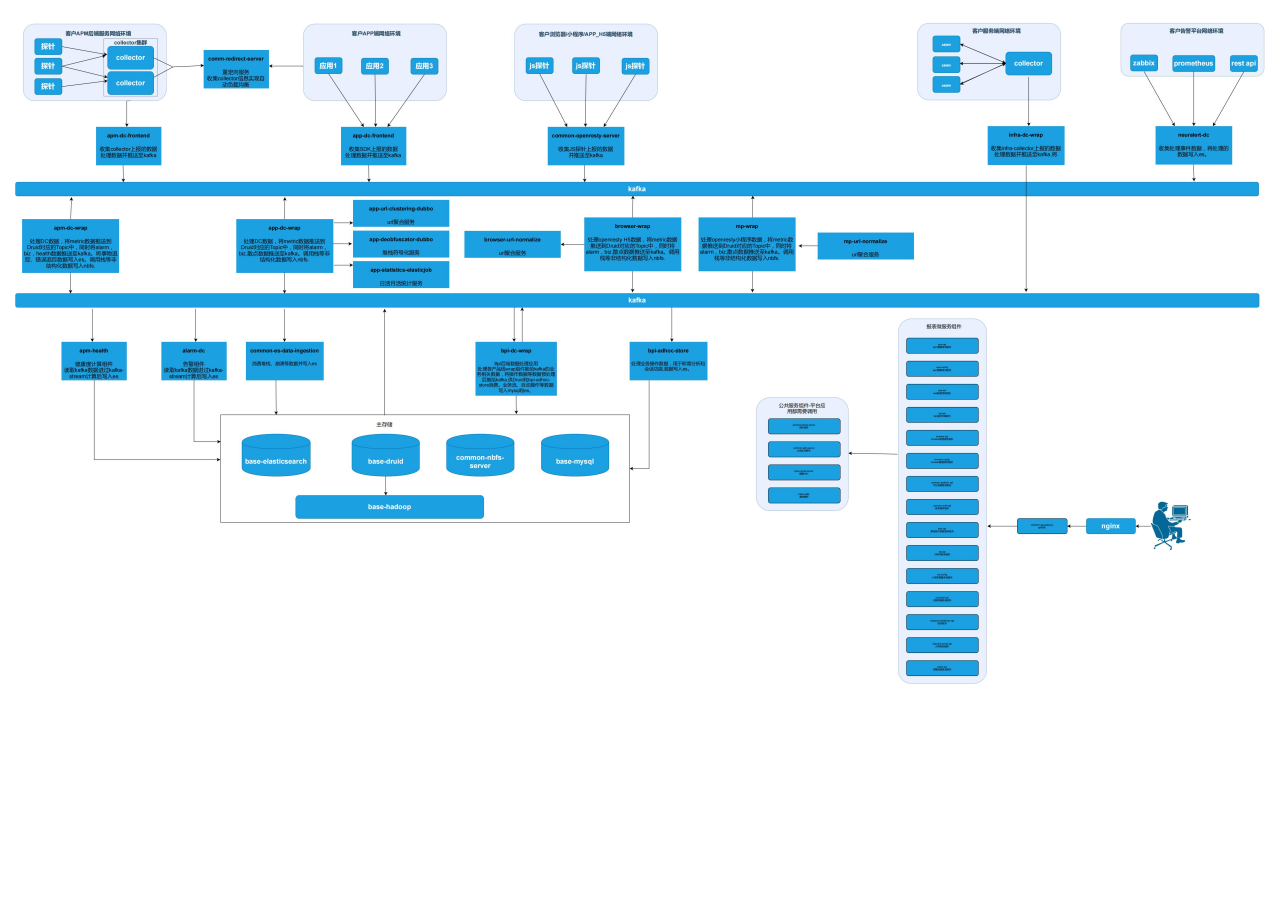


## 对外网络策略



|  |  |  |  |
| --- | --- | --- | --- |
| **源IP** | **目标IP** | **目标端口** | **策略说明** |
| 公网Internet | 行内公网LB(F5/NGINX） | 443(可根据行内要求调整) | app/browser真实用户数据采集 |
| 行内公网LB(F5/NGINX） | nginx-02/03 nginx | 8089 | app/browser数据转发 |
| 生产网络 | nginx-01 nginx | 8080/18848 | 平台页面访问&配置中心访问 |
| 生产网络 | druid-01 | 8888 | druid管理页面访问 |
| 行内业务被监控服务器 | collector-01/02 | 7665 7666 | apm&infra数据上报 |

## 逻辑架构图



# 环境说明

## 操作系统

操作系统版本要求：

基调听云要求的操作系统及版本如下：

| 系统 | 版本 | 备注 |
| --- | --- | --- |
| CentOS | 7.5-7.9 |  |
| RedHat | 7.5-7.9 |  |

系统分区：无特殊要求，数据盘挂载到单独目录。比如/data或者 /tingyun等均可。关闭swap分区。

用户规划：（行内若无特殊要求，使用下表。有特殊要求，依据要求规划即可）

|  |  |  |  |
| --- | --- | --- | --- |
| 组别 | 用户 | 家目录 | 说明 |
| tingyun | tingyun | /home/tingyun | 应用日常维护用户 |

## 基础环境

* 关闭selinux
* 关闭防火墙
* 关闭swap分区
* 设置时间同步服务器
* 可使用netstat ssh ssh-copy-id等命令
* 配置22台主机普通用户tingyun 的ssh免密
* 机器hostname 配置规范

# 软件安装与配置

## 机器初始化

#注意5.2章节基础环境要求

初始化操作需要在所有机器执行：

下文以tingyun用户，安装目录为/data/tingyun，服务器hostname为

apm01 apm02 apm03 ，对应ip为10.0.0.1 10.0.0.2 10.0.0.3为例

------------------------------------------------------------------------------------------------------#新增用户组 用户 安装目录并修改目录属主

groupadd tingyun

useradd -m tingyun -g tingyun

echo "68jVk@KD"|passwd tingyun --stdin

mkdir -p /data/tingyun

chown -R tingyun:tingyun /data/tingyun

#修改系统句柄数

cat >>/etc/security/limits.d/20-nproc.conf <<EOF

\* soft nproc 30000

root soft nproc unlimited

EOF

sed "s/^\\*(.\*)/#\\*\1/" -ri /etc/security/limits.conf

cat >>/etc/security/limits.conf <<EOF

\* hard nofile 1000000

\* soft nofile 1000000

\* hard nproc 819200

\* soft nproc 819200

EOF

echo "fs.file-max = 1000000" >>/etc/sysctl.conf

echo "vm.max\_map\_count = 262144" >>/etc/sysctl.conf

echo "vm.swappiness=10" >>/etc/sysctl.conf

echo "net.ipv4.tcp\_max\_tw\_buckets = 262144" >>/etc/sysctl.conf

sysctl -p >/dev/null 2>&1

#修改机器/etc/hosts

cp /etc/hosts /etc/hosts.tingyun.bak -a

cat >/etc/hosts <<EOF

127.0.0.1 localhost.localdomain localhost

127.0.0.1 localhost4.localdomain4 localhost4

::1 localhost.localdomain localhost

::1 localhost6.localdomain6 localhost6

#这里是配置所有服务器节点，而且是标准的 ip hostname一对一的方式

10.0.0.1 apm01

10.0.0.2 apm02

10.0.0.3 apm03

EOF

------------------------------------------------------------------------------------------------------

#初始化完毕

## 相关安装配置修改

选择一台机器为主节点，在该节点执行安装，该节点也就作为后续维护的管理节点。上传安装包V3.6.2.0-release\_20220113.zip，并解压

tar -xf V3.6.2.0\_release.tar.gz

for x in $(ls \*.tar.gz |grep -v V3.6.2.0-release);do tar -xf $x -C ./V3.6.2.0-release/apps/ ;done

# 安装部署配置文件修改

# 从 templates 目录拷贝对应安装环境的模版文件，此处以集群版 3 台服务器为例

#单机的就copy guard\_micro.cfg

## cp ../templates/guard\_micro\_3.cfg guard\_micro.cfg

cd V3.6.2.0\_release/script/install/

cp ../templates/guard\_cluster\_3.cfg guard.cfg

## 集群 vim guard.cfg 单机修改 guard\_micro.cfg

## 需修改 [hosts] 模块下的内容，以及 [hosts.\*] 的 值

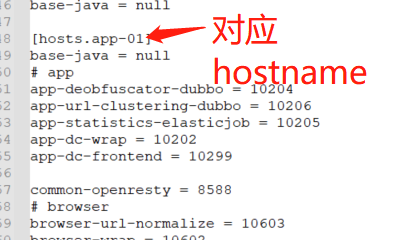
## 按各自环境修改相关配置，这一步由听云工程师现场修改或者远程提供修改后的文件

vim guard.cfg

本文22台主机分布配置文件如下，需要根据实际情况修改文件中的hostname ssh端口等。

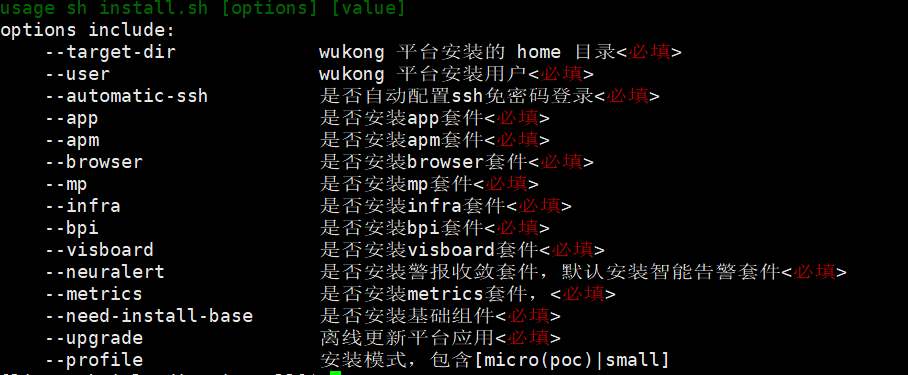






## 执行安装

## 将需要安装的产品线置为 true



sh install.sh --target-dir /data/tingyun --user tingyun --automatic-ssh false --app true --apm true --browser true --mp false --infra true --bpi true --visboard true --neuralert true --metrics true --need-install-base true --upgrade false --profile small

## 输出日志说明

## 主要日志输出路径,以下为示例目录

1) /var/tmp/tingyun\_install.log

2) /data/pkgs/v3.6.2.0\_release/script/install.log

3) /data/tingyun/package\_tmp/\*.log

## 部署验证

部署脚本执行完毕，在管理节点（执行部署脚本的节点即为管理节点）执行状态检查：

sh /data/tingyun/service\_manager.sh status all

验证组件运行状态，确定组件运行无误，继续执行6.6功能hotfix部署。如有问题，听云工程师进行处理。

## 功能hotfix部署

功能hotfix版本：V3.6.2.3

验证部署无误后，进行功能hotfix安装。

解压总包：3623-all-20220316.tar.gz

#解压安装包（${VersionName}为此次发版的版本号）

tar -xf ${VersionName}.tar.gz

#解压产品线安装包

tar -xf ${ProductName}.tar.gz -C ${versionName}/apps/

cd ${versionName}/script/

# 执行脚本

sh hotfix\_deploy.sh --target-dir /data/tingyun --operation-type update

## 额外nginx部署

部署报表页面查看和apm/app数据转发nginx。

只需要复制comon-01的nginx机器，然后修改对应配置文件即可。

Common-01: scp -r base-nginx 目标主机（nginx/collector）:/data/tingyun/base

然后到目标主机:vim /data/tingyun/base/base-nginx/conf/nginx.conf即可